logo

[Global Technical Committee]

FIXML Technical Standard Version 1.2

2016-04-20-

Rev 0.3

Proposal Status: Draft

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE "FIX PROTOCOL") ARE PROVIDED "AS IS" AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

**DRAFT OR NOT RATIFIED PROPOSALS** (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FPL GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”. THE FPL GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Copyright 2003-2015 FIX Protocol Limited, all rights reserved.

Table of Contents

[Document History 5](#_Toc448913096)

[1 Introduction 6](#_Toc448913097)

[1.1 Authors 6](#_Toc448913098)

[2 Business and Technical Requirements 6](#_Toc448913099)

[2.1 Encoded Fields Support 6](#_Toc448913100)

[2.2 Binary Data Content Support 7](#_Toc448913101)

[2.3 FIXML Inlined Components Changes 7](#_Toc448913102)

[2.4 FIX XML Namespace Standard 7](#_Toc448913103)

[3 Issues and Discussion Points 7](#_Toc448913104)

[3.1 Should base64Binary encoding be used? 7](#_Toc448913105)

[3.2 Should encoded data be stored in elements or attributes? 8](#_Toc448913106)

[3.3 Should the related data length field be included in FIXML? 8](#_Toc448913107)

[3.4 Should the encoded data fields be extended to include an IETF standard content type? 10](#_Toc448913108)

[3.5 Are current FIXML fields of type data affected by these changes? 10](#_Toc448913109)

[3.6 Should fields of type NumInGroup also be included as attributes? 11](#_Toc448913110)

[4 References 12](#_Toc448913111)

[5 Relevant and Related Standards 12](#_Toc448913112)

[6 Intellectual Property Disclosure 12](#_Toc448913113)

[7 Definitions 12](#_Toc448913114)

[8 Technical Changes and Enhancements 13](#_Toc448913115)

[8.1 FIXML Datatype Changes 13](#_Toc448913116)

[8.1.1 Change of data XML datatype (xs:base64Binary) 13](#_Toc448913117)

[8.1.2 Revise Length XML datatype attributes 14](#_Toc448913118)

[8.1.3 Revise NumInGroup XML datatype attributes 14](#_Toc448913119)

[8.2 FIXML Patterns for Fields of type data 15](#_Toc448913120)

[8.2.1 Alternative 1 – Field references (for data fields) as attributes 15](#_Toc448913121)

[8.2.2 Alternative 1 - Fields of type data defined as simpleType 16](#_Toc448913122)

[8.2.3 Alternative 2 - Field references (for data fields) as elements 16](#_Toc448913123)

[8.2.4 Alternative 2 - Fields of type data defined as complexType 17](#_Toc448913124)

[8.2.5 Field references (for Length fields) are not included as attributes 18](#_Toc448913126)

[8.3 FIXML Namespaces 19](#_Toc448913130)

[Appendix A - Usage Examples 19](#_Toc448913131)

[Alternative 1 – Field references implemented as attributes 20](#_Toc448913132)

[Alternative 2 – Field references implemented as elements 21](#_Toc448913133)

[Appendix B – Compliance Strategy 22](#_Toc448913134)

Table of Tables

[Table 1 - Length fields incorrectly included in FIXML 8](#_Toc448913135)

[Table 2 - fields of type data in FIXML 10](#_Toc448913136)

[Table 3 - data Datatype 13](#_Toc448913137)

[Table 4 - Length Datatype 14](#_Toc448913138)

[Table 5 - NumInGroup Datatype 14](#_Toc448913139)

# Document History

| **Revision** | **Date** | **Author** | **Revision Comments** |
| --- | --- | --- | --- |
| 0 | Dec 31, 2014 | Jim Northey | Initial draft. |
|  | Feb 15, 2015 | Rich Shriver | Revised business requirements to align with the needs of EP197. |
|  | May 25, 2015 | Rich Shriver | Drafted section 8 to align with prototyping work completed with Jim Northey. |
|  | July 6, 2015 | Rich Shriver | Completed initial revision of Section 8 to identify the FIXML implementation details. |
|  | August 7, 2015 | Jim Northey | Added content to discussion section and other minor edits. |
|  | August 28, 2015 | Rich Shriver | Added sections to include NumInGroup fields in the FIXML schema as optional attributes.  Added sections to include removal of the OptimisedImplicitBlockRepeating component type mapping to FIXML. |
| 0.2 | December 21, 201 | Jim Northey | Added background on the inclusion of NumInGroup fields. |
| 0.3 | February 25, 2016  April 20, 2016 | Rich Shriver  Rich Shriver, Don Mendelson, Jim Northey | Incorporated feedback from Andy Key and Ryan Pierce. Revised Section 3 with their comments and feedback. Removed several features from Section 8 including support for Length and NumInGroup fields, Mime encoding and added an alternative for data fields to be included as an attribute  Included options for data fields in FIXML as elements and attributes and the discussion to date of this decision.  Added changes to conform to the proposed FIX XML namespace standard. |
|  |  |  |  |
|  |  |  |  |

# Introduction

To meet new requirements for FIXML this technical standard proposal calls for FIXML, the XML encoding of FIX, to be extended to include all encoded data fields with content of type base64binary. This proposal includes the FIXML Inline Component Reference proposal implemented to support EP105.

## Authors

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Affiliation** | **Contact** | **Role** |
| Jim Northey | Itivity AB | [jim.northey@itivity.com](mailto:jim.northey@itivity.com ) | GTC Co-chair and co-author |
| Rich Shriver | R. Shriver Associates LLC | [rich@rshriver.com](mailto:rich@rshriver.com) | Co-author |
|  |  |  |  |
|  |  |  |  |

# Business and Technical Requirements

The FIX datatype of data was created in FIX 4.2 as part of the technique to support the exchange of binary data where there is a requirement for content to be able to include reserved values of <SOH> 0x01, which is the standard delimiter in the FIX tag=value encoding.

For tag=value encoding, the technique requires two fields to be in sequence – the length of the raw data field and the raw data field. The length of the raw data field is used to determine the raw data field boundary with a datatype of Length. The raw data field contains the binary (unrestricted) data content and is of datatype data.

The data datatype is a string field containing raw data with no format or content restrictions.

Several fields in FIX such as Issuer(106) have related encoded fields – EncodedIssuerLen(348) and EncodedIssuer(349). The EncodedIssuerLen(348) field value is used to provide the FIX tag=value decoder with information to read the encoded data, as a search for the delimiter (0x01) would not produce reliable results.

Many of the encoded fields have not been included in FIXML and XML content does not have the same restriction as the tag=value encoding. An incremental improvement opportunity exists to provide a means for including binary data content within FIXML.

## Encoded Fields Support

Work complete by staff at Standard Bank in London in 2008 identified the need to include the encoded fields within FIXML. In markets where encoded data fields are used, it is often the case that the unencoded fields are also included in the FIX messages. The lack of complete support for encoded data fields in FIXML means that there is information loss when converting between FIX tag=value encoded data and FIXML.

## Binary Data Content Support

Recently, as part of the extension of FIX to support CFTC Parts 43 – 45 regulations (EP197), there are new requirements with Equity Swaps to include an encoded clip image of a payment formula within FIXML. This formula image is natively encoded using base64 binary data.

The recommendation to better support encoded fields is to use the xs:base64Binary XML Schema datatype for encoded data fields. A survey of other standards, such as FpML, the base64Binary datatype is already an established practice for encoding binary data within XML documents.

## FIXML Inlined Components Changes

The proposal for inlined FIXML components (FIXML Inline Component Reference 0.2) was implemented in EP105 to provide greater efficiency, coverage and flexibility in the FIXML schema generation and was intended to replace the need for the legacy OptimizedImplicitBlockRepeating component type and its associated FIXML mapping. Two (2) components (InstrmtLegGrp and UndInstrmtGrp) were identified as temporary issues for FPL constituents and their revision was postponed. In EP192, the addition of LegFinancingDetails to the InstrmtLegGrp component provided a catalyst for changing the two remaining OptimisedImplicitBlockRepeating components to BlockRepeating components with inlined component references. This proposal includes the removal of FIXML code generation for components of the type OptimisedImplicitBlockRepeating. The "Optimized" concept is no longer supported in FIXML.

## FIX XML Namespace Standard

The formal namespace for FIX Protocol is being changed from fixprotocol.org to fixprotocol.io. The FIX XML Technical Standard Proposal is for all FIX XML Schemas. Namespace attributes in the FIXML schema are affected and new attributes are being proposed to identify the FIX version, extension pack and custom version (for use in implementation). The figure below illustrates the proposed change to the schema element including the change in namespace and the addition of version attributes.

# Issues and Discussion Points

The following is a summary list of issues and discussion points which are explored in greater detail in this section below.

1. Should base64binary encoding be used?
2. Should encoded data be stored in elements or attributes?
3. Should the data length field be included in FIXML?
4. Should the encoded data fields be extended to include an IETF standard content type?
5. Are current FIXML fields of type data affected by these changes?
6. Should fields of type NumInGroup be included in FIXML as attributes?

## Should base64Binary encoding be used?

The use of base64Binary encoding is a widely used technique for capturing large quantities of information within an XML document. Base64Binary is used within FpML. A potential drawback is that the requirement to use base64Binary will require a conversion routine when transforming between FIXML and other FIX encoding types that are not using base64Binary encoding.

## Should encoded data be stored in elements or attributes?

There have been several discussions on whether the data fields should be included in FIXML as elements or attributes. Some would propose the best practice is to use attributes for fields of generally smaller content and elements for fields where the size of data may be quite large. As an example of the new business requirements, a binary image of a formula may be considered a relatively large amount of data. There are no technical limits in XML to the size of either elements or attributes. An external discussion of this can be found here (<http://stackoverflow.com/questions/1334776/how-many-characters-allowed-in-xml-attribute>).

Others propose that there is not a best practice and clearly no standard for determining whether a field should be implemented as an element or an attribute in XML. Largely because the current model for FIXML is to include fields as attributes and components as elements, this proposal recommends preserving the original design and not introducing a change to incorporate the fields of type data as elements in FIXML.

## Should the related data length field be included in FIXML?

In XML, the length attribute is not required for low level data parsing as it is in the FIX tag=value encoding. The length value is not needed in FIXML to determine the field termination as it is enclosed within XML tags. Fields of type Length are structural elements particular to some encodings, not general purpose semantics. Further, should the length value not match the length of the data for some reason, it is not clear how to best handle the discrepancy. At this point, without compelling reasons, the length attribute (in the repository and critical for tag=value syntax) is not proposed for inclusion in FIXML and rather, those length fields that have been coded with incorrect values that result in Length fields being added to the FIXML attribute groups should be corrected in the repository to indicate the field is not required in FIXML. The FIXML generation code should be revised to ignore fields and field references of type Length in the generation of the FIXML schema. A list of these fields with incorrectly coded NotReqFIXML is included in the table below.

Table 1 - Length fields incorrectly included in FIXML

| Tag | Name |
| --- | --- |
| 95 | RawDataLength |
| 348 | EncodedIssuerLen |
| 350 | EncodedSecurityDescLen |
| 352 | EncodedListExecInstLen |
| 354 | EncodedTextLen |
| 356 | EncodedSubjectLen |
| 358 | EncodedHeadlineLen |
| 360 | EncodedAllocTextLen |
| 362 | EncodedUnderlyingIssuerLen |
| 364 | EncodedUnderlyingSecurityDescLen |
| 445 | EncodedListStatusTextLen |
| 618 | EncodedLegIssuerLen |
| 621 | EncodedLegSecurityDescLen |
| 1277 | DerivativeEncodedIssuerLen |
| 1280 | DerivativeEncodedSecurityDescLen |
| 1397 | EncodedMktSegmDescLen |
| 1401 | EncryptedPasswordLen |
| 1403 | EncryptedNewPasswordLen |
| 1664 | EncodedRejectTextLen |
| 1578 | EncodedEventTextLen |
| 2072 | EncodedUnderlyingEventTextLen |
| 2074 | EncodedLegEventTextLen |
| 40004 | EncodedAdditionalTermBondDescLen |
| 40008 | EncodedAdditionalTermBondIssuerLen |
| 40978 | EncodedLegStreamTextLen |
| 40980 | EncodedLegProvisionTextLen |
| 40982 | EncodedStreamTextLen |
| 40984 | EncodedPaymentTextLen |
| 40986 | EncodedProvisionTextLen |
| 40988 | EncodedUnderlyingStreamTextLen |
| 2111 | EncodedAttachmentLen |
| 41083 | EncodedDeliveryStreamCycleDescLen |
| 1525 | EncodedDocumentationTextLen |
| 1678 | EncodedOptionExpirationDescLen |
| 2179 | EncodedLegOptionExpirationDescLen |
| 41101 | EncodedMarketDisruptionFallbackUnderlierSecurityDescLen |
| 41107 | EncodedExerciseDescLen |
| 41256 | EncodedStreamCommodityDescLen |
| 41320 | EncodedLegAdditionalTermBondDescLen |
| 41324 | EncodedLegAdditionalTermBondIssuerLen |
| 41458 | EncodedLegDeliveryStreamCycleDescLen |
| 41476 | EncodedLegMarketDisruptionFallbackUnderlierSecurityDescLen |
| 41482 | EncodedLegExerciseDescLen |
| 41653 | EncodedLegStreamCommodityDescLen |
| 41806 | EncodedUnderlyingDeliveryStreamCycleDescLen |
| 41811 | EncodedUnderlyingExerciseDescLen |
| 2287 | EncodedUnderlyingOptionExpirationDescLen |
| 41873 | EncodedUnderlyingMarketDisruptionFallbackUnderlierSecurityDescLen |
| 41969 | EncodedUnderlyingStreamCommodityDescLen |
| 2372 | EncodedTradeContinuationTextLen |
| 2351 | EncodedComplianceTextLen |
| 41710 | EncodedUnderlyingAdditionalTermBondDescLen |
| 42025 | EncodedUnderlyingAdditionalTermBondIssuerLen |
| 42171 | EncodedUnderlyingProvisionTextLen |
| 2481 | EncodedMDStatisticDescLen |
| 2494 | EncodedLegDocumentationTextLen |
| 2522 | EncodedWarningTextLen |
| 2637 | EncodedMiscFeeSubTypeDescLen |
| 2651 | EncodedCommissionDescLen |
| 2665 | EncodedAllocCommissionDescLen |
| 42451 | LegPaymentStreamFormulaImageLength |
| 42652 | PaymentStreamFormulaImageLength |
| 42947 | UnderlyingPaymentStreamFormulaImageLength |

## Should the encoded data fields be extended to include an IETF standard content type?

The current proposal raises the possibility of using an extension to XML that specifies a mime type. The advantage of this is to provide meta data for message consumers to be able to more intelligently process payload. While this does not sound like a bad approach, there is no equivalent mechanism available in other FIX encodings, such as FIX tag=value. This proposal does not include the addition of mime types as this is more appropriately considered at a broader level.

## Are current FIXML fields of type data affected by these changes?

Some of the fields of type data are currently included in FIXML with a data type of data which is restricted by xs:string. These fields and the associated length fields are included as component attributes if they include the field attribute NotReqFIXML = “0”. The table below lists all fields of type data and indicates whether or not these fields are included in FIXML. Since this proposal is recommending inclusion of fields of type data, the field attribute of NotReqFIXML = “1” (Fields NOT currently included in FIXML) should be changed to reflect NotReqFIXML = “0” and should be included as component attributes in FIXML.

Table 2 - fields of type data in FIXML

| Fields that are currently included in FIXML | Fields NOT currently included in FIXML |
| --- | --- |
| RawData(96) | Signature(89) |
| EncodedIssuer(349) | SecureData(91) |
| EncodedSecurityDesc(351) | XmlData(213) |
| EncodedListExecInst(353) | EncodedText(355) |
| EncodedSubject(357) | DerivativeSecurityXML(1283) |
| EncodedHeadline(359) | EncodedSecurityListDesc(1469) |
| EncodedAllocText(361) | InstrumentScopeEncodedSecurityDesc(1621) |
| EncodedUnderlyingIssuer(363) | EncodedRejectText(1665) |
| EncodedUnderlyingSecurityDesc(365) | EncodedFirmAllocText(1734) |
| EncodedListStatusText(446) | EncodedEventText(1579) |
| EncodedLegIssuer(619) | EncodedUnderlyingEventText(2073) |
| EncodedLegSecurityDesc(622) | EncodedLegEventText(2075) |
| DerivativeEncodedIssuer(1278) | EncodedDocumentationText(1527) |
| DerivativeEncodedSecurityDesc(1281) | EncodedOptionExpirationDesc(1679) |
| EncodedMktSegmDesc(1398) | EncodedLegOptionExpirationDesc(2180) |
| EncryptedPassword(1402) | EncodedUnderlyingOptionExpirationDesc(2288) |
| EncryptedNewPassword(1404) | EncodedLegDocumentationText(2493) |
| EncodedAttachment(2112) | EncodedWarningText(2521) |
| EncodedTradeContinuationText(2371) | EncodedAdditionalTermBondDesc(40005) |
| EncodedComplianceText(2352) | EncodedAdditionalTermBondIssuer(40009) |
| EncodedMDStatisticDesc(2482) | EncodedLegStreamText(40979) |
|  | EncodedLegProvisionText(40981) |
|  | EncodedStreamText(40983) |
|  | EncodedPaymentText(40985) |
|  | EncodedProvisionText(40987) |
|  | EncodedUnderlyingStreamText(40989) |
|  | EncodedDeliveryStreamCycleDesc(41084) |
|  | EncodedMarketDisruptionFallbackUnderlierSecurityDesc(41102) |
|  | EncodedExerciseDesc(41108) |
|  | EncodedStreamCommodityDesc(41257) |
|  | EncodedLegAdditionalTermBondDesc(41321) |
|  | EncodedLegAdditionalTermBondIssuer(41325) |
|  | EncodedLegDeliveryStreamCycleDesc(42459) |
|  | EncodedLegMarketDisruptionFallbackUnderlierSecurityDesc(41477) |
|  | EncodedLegExerciseDesc(41483) |
|  | EncodedLegStreamCommodityDesc(41654) |
|  | EncodedUnderlyingDeliveryStreamCycleDesc(41807) |
|  | EncodedUnderlyingExerciseDesc(41812) |
|  | EncodedUnderlyingMarketDisruptionFallbackUnderlierSecurityDesc(41874) |
|  | EncodedUnderlyingStreamCommodityDesc(41970) |
|  | EncodedUnderlyingAdditionalTermBondDesc(41711) |
|  | EncodedUnderlyingAdditionalTermBondIssuer(42026) |
|  | EncodedUnderlyingProvisionText(42172) |

## Should fields of type NumInGroup also be included as attributes?

In discussion, it had been considered to include NumInGroup fields within the block repeating components to support translation from FIXML to another encoding However, this information is available from XML and is not required to interpret the FIXML data. Fields of type NumInGroup are structural elements particular to some encodings, not general purpose semantics. Further, it is not clear how to handle a situation where the value in the field does not match the actual number of entries in the group. Without the identification of compelling requirements, this proposal does not recommend inclusion of the fields of type NumInGroup in the FIXML schema.

Presently, no fields of the datatype NumInGroup are generated in the FIXML schema. Some of the fields include an abbreviated name and most, but not all of the fields are encoded with the attribute NotReqXML = 1 in the FIX Repository. This meta data may need attention and cleanup in support of future implementations.

# References

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference** | **Version** | **Relevance** | **Normative** |
| XML Schema Part 0: Primer Second Edition  [http://www.w3.org/TR/xmlschema-0/#DefnDeclars](http://www.w3.org/TR/xmlschema-0/%23DefnDeclars) |  | Provides background on declaring simple and complex types | Incorporates |
| Describing Media Content of Binary Data in XML  [http://www.w3.org/TR/xml-media-types/#contentType](http://www.w3.org/TR/xml-media-types/%23contentType) |  | Recommended approach for specifying MIME types for binary data transmitted within an XML element. | Incorporates |
|  |  |  |  |
|  |  |  |  |

# Relevant and Related Standards

|  |  |  |  |
| --- | --- | --- | --- |
| **Related Standard** | **Reference location** | **Relationship** | **Normative** |
| XML Schema |  | Uses | Yes |
| Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies | <https://tools.ietf.org/html/rfc2045> |  | No |
| Multipurpose Internet Mail  (MIME) Part Two: Media Types | <https://tools.ietf.org/html/rfc2046> | Uses | Yes |
| FIXML Specification 1.0 | Not yet published | Extends | Yes |
| FpML |  | References | No |
| FIX XML Namespaces | Not yet published | Uses |  |
|  |  |  |  |

# Intellectual Property Disclosure

There are no intellectual property issues identified at this time.

# Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |

# Technical Changes and Enhancements

This section identifies the technical changes and enhancement to FIXML namespaces, datatypes and mapping patterns for support of encoded data fields.

## FIXML Datatype Changes

The proposal includes recommendations for changes to two (2) datatypes (data and NumInGroup). The XML mapping of the data datatype is recommended to be changed from xs:String to xs:base64Binary. The Length datatype is also being recommended for a minor revision to the XML Description to indicate that fields of this type should not be included in FIXML. Details of these changes are included in the subsections below.

### Change of data XML datatype (xs:base64Binary)

This proposal is recommending the change of the xml datatype attribute of the FIX data datatype to be changed from xs:string to xs:base64Binary. The table below lists the current information in the FIX Repository for the datatype of data and the recommended changes to the datatype metadata. The XML base type is recommended to change from xs:string to xs:base64Binary. The recommendation is to identify this datatype as an XML builtin datatype (set value = "1"). The data type description is also recommended to be replaced with a description that indicates all fields of type data are to be encoded using the base64Binary encoding.

Table 3 - data Datatype

|  |  |
| --- | --- |
| **Type name** | data |
| **Base type** | String |
| **Description** | string field containing raw data with no format or content restrictions. Data fields are always immediately preceded by a length field. The length field should specify the number of bytes of the value of the data field (up to but not including the terminating SOH).  Caution: the value of one of these fields may contain the delimiter (SOH) character. Note that the value specified for this field should be followed by the delimiter (SOH) character as all fields are terminated with an "SOH". |
| **XML builtin** | 1 (builtin) |
| **XML base type** | xs:base64Binary |
| **XML minInclusive** |  |
| **XML pattern** |  |
| **XML Description** | base64Binary In FIXML, all data type fields are encoded using base64Binary encoding. |

**Example:**

The FIXML shema for the data datatype should appear as follows:

<xs:simpleType name="data">

<xs:annotation>

<xs:documentation>base64Binary In FIXML, all data type fields are encoded using base64Binary encoding.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:base64Binary"/>

</xs:simpleType>

### Revise Length XML datatype attributes

A note is proposed to be added to the Length XML Description to indicate Length fields should not be included in FIXML.

Table 4 - Length Datatype

|  |  |
| --- | --- |
| **Type name** | Length |
| **Base type** | int |
| **Description** | int field representing the length in bytes. Value must be positive. |
| **XML builtin** | ~~0~~ |
| **XML base type** | ~~xs:nonNegativeInteger~~ |
| **XML minInclusive** |  |
| **XML pattern** |  |
| **XML Description** | ~~int field representing the length in bytes. Value must be positive.~~  Note - Length fields should not be included in FIXML. |

The FIXML generation code should be changed to ignore fields and field references where the datatype is Length.

### Revise NumInGroup XML datatype attributes

A note is proposed to be added to the NumInGroup XML Description to indicate fields of this type are not included in FIXML.

Table 5 - NumInGroup Datatype

|  |  |
| --- | --- |
| **Type name** | NumInGroup |
| **Base type** | Int |
| **Description** | int field representing the number of entries in a repeating group. Value must be positive inclusive of zero. |
| **XML builtin** | ~~0~~ |
| **XML base type** | ~~xs:nonNegativeInteger~~ |
| **XML minInclusive** |  |
| **XML pattern** |  |
| **XML Description** | ~~int field representing the number of entries in a repeating group. Value must be positive inclusive of zero.~~  Note – NumInGroup fields are not included in FIXML. |

The FIXML generation code should be changed to ignore fields and field references where the datatype is NumInGroup.

## FIXML Patterns for Fields of type data

Given the size of the image value, the frequent encoding of base64 data with 76 byte lines to support Internet email standards, the recommendation is to place the data within an element. This breaks with FIXML, which controversially stores data within attributes. The use of an element in this case permits us to include metadata as attributes to the encoded data field. One piece of metadata that can be included is the length of the field to assist in mapping between FIX tag=value and FIXML.

This proposal currently recommends the alternative 1 FIXML implementation. The two alternatives include:

* Alternative 1 – FIXML field references for fields of type data as attributes - Fields of type data defined as simpleType with base64Binary restriction.
* Alternative 2 – FIXML field references for fields of type data as implemented in FIXML as elements - fields of type data defined as complexType for base64Binary content

In both cases, Field references for fields of type Length are not included in the component or message attribute group.

### Alternative 1 – Field references (for data fields) as attributes

FIX Repository field references for fields coded with the datatype "data" are included in FIXML as attribute in the corresponding message or component attribute group.

* The attribute’s name is the field's abbreviated name (as found in the FIX Repository).
* The attribute’s type attribute is set to the data field simpleType (described below). The type name is a concatenation of the field name and "\_t" ("EncodedIssuer\_t" in the examples).
* For the use attribute, if the field reference is coded in the FIX repository as not required, the use attribute is set to "optional"; otherwise the use attribute is set to "required" for required fields.

In the example below, the field EncodedIssuer is included within the InstrumentAttributes attribute group.

**Example:**

<xs:attributeGroup name="InstrumentAttributes">

<< truncated>>

<xs:attribute name="Issr" type="Issuer\_t" use="optional"/>

<xs:attribute name="EncIssr" type="EncodedIssuer\_t" use="optional"/>

<xs:attribute name="Desc" type="SecurityDesc\_t" use="optional"/>

<xs:attribute name="EncSecDesc" type="EncodedSecurityDesc\_t" use="optional"/>

<< truncated>>

</xs:attributeGroup>

This attribute is included in FIXML when the notReqXML attribute of the field reference and the field in the FIX Repository are both set to "0". The attribute is not included in FIXML if the field reference or the data attribute in the FIX Repository for notReqXML is set to "1".

### Alternative 1 - Fields of type data defined as simpleType

FIX Repository fields coded with a type "data" are included in FIXML as simpleType declarations.

* The simpleType name attribute is set to the field name appended with the token "\_t" to identify the field type.
* The annotation of the simpleType is the same as is included for all FIX Repository fields.
* The restriction of the simpleType is set to “xs:base64Binary”.

The example below is for the simpleType definition of the EncodedIssuer(349) field.

**Example:**

<xs:simpleType name="EncodedIssuer\_t">

<xs:annotation>

<xs:documentation>Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Issuer field.</xs:documentation>

<xs:appinfo>

<fm:Xref Protocol="FIX" name="EncodedIssuer" ComponentType="Field" Tag="349" Type="data" AbbrName="EncIssr"/>

</xs:appinfo>

</xs:annotation>

<xs:restriction base="xs:base64Binary"/>

</xs:simpleType>

This simpleType is included in FIXML when the notReqXML attribute of the field in the FIX Repository is set to "0"; otherwise this simpleType is not included in FIXML when the notReqXML attribute of the field in the FIX Repository is set to "1".

### Alternative 2 - Field references (for data fields) as elements

FIX Repository field references for fields coded with the datatype "data" are included in FIXML as elements related to their corresponding message or component.

* The element’s name attribute is the field's abbreviated name.
* The element’s type attribute is set to the data field complexType (described below). The type name is a concatenation of the field name and "\_t" ("EncodedIssuer\_t" in the examples).
* For the minOccurs attribute, if the field reference is coded in the repository as not required, the element’s minOccurs attribute of the element is set to "0"; otherwise the element’s minOccurs attribute is set to "1" for required fields.
* The element’s maxOccurs value of the element is always set to "1".

In the example below, the field EncodedIssuer is included within the InstrumentEmelements group.

**Example:**

<xs:group name="InstrumentElements">

<xs:sequence>

<< truncated>>

<xs:element name="EncIssr" type="EncodedIssuer\_t" minOccurs="0" maxOccurs="1"/>

<< truncated>>

</xs:sequence>

</xs:group>

This element is included in FIXML when the notReqXML attribute of the field reference and the field are both set to "0". The element is not generated if the field reference or the data attribute notReqXML is set to "1".

### Alternative 2 - Fields of type data defined as complexType

FIX Repository fields coded with a type "data" are included in FIXML as complexType declarations.

* The name attribute is set to the field name appended with the token "\_t" to identify the field type.
* The annotation of the complexType is the same as is included for all FIX Repository fields.
* The type is simpleContent extended by the "data" datatype which when configured as described above, results in a base64Binary content.
* The optional Length attribute and xmime contentType are described in sections below.

The example below is for the complexType definition of the EncodedIssuer(349) field.

**Example:**

<xs:complexType name="EncodedIssuer\_t">

<xs:annotation>

<xs:documentation>Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Issuer field.</xs:documentation>

<xs:appinfo>

<fm:Xref Protocol="FIX" name="EncodedIssuer" ComponentType="Field" Tag="349" Type="data" AbbrName="EncIssr"/>

</xs:appinfo>

</xs:annotation>

<xs:simpleContent>

<xs:extension base="data">

<xs:attribute name="EncIssrLen" type="EncodedIssuerLen\_t"/>

<xs:attribute ref="xmime:contentType" use="optional"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

This complexType is included in FIXML when the notReqXML attribute of the field is set to "0"; otherwise this complexType is not included in FIXML when the notReqXML attribute of the field in the FIX Repository is set to "1".

### Field references (for Length fields) are not included as attributes

The Length field reference should not be included in the attribute group along with other field references for a component or message (as they exist in FIXML 1.1). This is an repository data issue as well as a FIXML generation concern.

## FIXML Namespaces

FIXML Namespaces will follow the FIX XML Schema Technical Standard as proposed.

The Namespace convention is to be <http://www.fixprotocol.io/2004/fixml>. The FIXML schema in its current form was originally released in 2004.

The Namespace qualification is consistent with the FIX XML Schema Technical Standard (elementFormDefault="qualified" attributeFormDefault="unqualified").

The copyright and license notice will change to comply with the FIX XML Schema Technical Standard.

The FIX version metadata attributes (fixVersion and fixExtensionpack) will be included with each new FIXML schema for a FIX release and extension pack.

Copyright and license information will conform to the proposed FIX XML Schema Technical Standard.

The xsi:schemaLocation attribute will be removed from the released FIXML schema. The FIXML schema should not be served but rather information about the schema should be available and information about the FIXML schema will be available at the namespace url.

An example below illustrates the new FIXML schema element and the copyright and license notification.

**Example:**

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="http://www.fixprotocol.io/2004/fixml"

xmlns:fm=”http://www.fixprotocol.io/2004/fixml/METADATA”

xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”

targetNamespace="http://www.fixprotocol.io/2004/fixml"

elementFormDefault="qualified" attributeFormDefault="unqualified"

fixVersion="FIX5.0SP2” fixExtensionpack="EP208”>

<xs:annotation>

<xs:documentation>

© Copyright 2016 FIX Protocol Limited

Creative Commons Attribution-NoDerivatives 4.0

International Public License

</xs:documentation>

</xs:annotation>

</xs:schema>

# Appendix A - Usage Examples

The following are examples of FIXML documents using the EncodedIssuer field. The EncodedIssuer field includes base64Binary encoded data, within the Instrument component. Alternative 1 illustrates inclusion of the field reference as an attribute and Alternative 2 illustrates inclusion of the field reference as an element.

## Alternative 1 – Field references implemented as attributes

The following is an example of a FIXML file with the field reference EncIssr included in the Instrmt component of an Order as an attribute.

<?xml version="1.0" encoding="UTF-8"?>

<FIXML xmlns="http://www.fixprotocol.io/2004/fixml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" fixVersion="FIX.5.0SP2" fixExtensionpack="208">

<Order ID="123456" Side="2" TxnTm="2001-09-11T09:30:47-05:00" Typ="2" Px="93.25" Acct="26522154">

<Instrmt Sym="IBM" ID="459200101" Src="1" EncIssr="/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcG

BwcICQsJCAgKCAcHCg0KCgsMDAwMBwkODw0MDgsMDAz/2wBDAQICAgMDAwYDAwYMCAcIDAwMDAwM

DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAARCAApAGEDASIA

AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA

AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3

ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm

p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA

AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx

BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK

U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3

uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9/KKK

/PDSv+DjDwj4z8ReJrLwd+zL+2R8QrTwnrl34evtV8KfDqHVtON3bSFJUWaK8Iz0OGw21lJUZpcy

5uTra/yVl+q+8dny83Tb8/8AJn6H0V5f+yF+07/w1z8G4fGP/Cvfih8MfOu5rX+w/H+g/wBi6zH5

ZA8xrfe+I2zlW3cgGvUKqUXF2ZMZJq6CiiikMKKKKACiiigAorgvjr8ALH9oKw0yx1bxH480XStP

ne4mtfDHiW78PPqLFCqeddWTxXYRMlgkc6IxPzhwFA8D/wCCOvjbxP4z+DHxKXU/E+u+NfBOi/Er

XNJ+H2va1dyX97qOgwSIkZa7kLSXaJci6jSeRnd0jXLtgUQ96UodUub5JxX33louqTd9LBP3YqXR

u3zak/utF697K3U+uaKKKACvw+/4I6f8FQv+GN/BPx08K/8ADO/7UnxV+0fGXxLqH9sfDvwF/bml

R75o18hp/PjxMuzLJt4Doc81+4NfjF/wS/8A24PF3/BM/wANfGTwT4x/ZN/bI8U3eu/FbxB4ksdR

8KfDCa+064tLiVFiZZZZYSxPlFsqpUqykMc1FN2xD1tenJevv0nb8L/IdRXorS9pxfp7lRX/ABt8

z9bfgD8Xv+F+fBrw94x/4Rfxh4L/AOEgtBdf2H4q03+ztZ03JI8u5t9zeVJxnbuPBFefftv/APDS

n/CNaF/wzb/wo7+2PtUn9tf8LK/tT7N5G0eX9n+wfN5m7O7fxjGOa2/2Of2qP+GwvhCfF3/CuPir

8LcX8tj/AGL8QvD/APYmrnywh87yPMf902/Ctu5KtxxWJ+2/+xD/AMNt+GtC03/hb3xx+EX9hXUl

z9q+Gvir+wLnUN6hfLuH8qTzEXGVXAwSTV4mLv7qtqnv0un+X37CoSVtXfdbddV+Z85R/wDD0vzF

3f8ADAW3PzY/4S7OPavvu33+Qnm7fM2jfs+7nvj2r4Ej/wCCB3lyK3/Daf7fpwc4Pxe4P1/0Svvu

3i+zwIm532KF3Ocs2O5PrWl1yJdb/wCXUnXmufl58X/2n/2u/jj/AMFkfjF+zr8G/G/hXwh4M0jw

7omqSeJNZ0S11GTwPFLAGmls7baj3tzPK6BVuJGhRUfhSykcV8Nv2k/299O/bn8ffsbP8Rvhr4w8

TwaZa+K7T4y6n4bgsbnw/ocwEUrpo9vi3uLoTOghjc7VZJPNaRJFMX0J+zF8HfF2gf8ABwR+0x4z

vvC3iOy8H694E8O2ema7Ppk0em6jPEkfmxQ3BURyOmDuVWJXHIFN+G/wd8XWP/ByF8RvHM3hXxHD

4JvvgzY6XbeIH0yZdKuLtb+N2tkuSvlNKFBYoG3ADOKwwUbwoKevOqnNf+77WUdd1rCKumm07bNW

rFSadVx+z7Ll+apKXr8Ur3urq+9zI/4J/ftRfHz4M/8ABT3xx+yl8fvH+l/GOb/hD4fH/hPxtbeG

rfw/cy2xljtprOe1tv3ICybypG5vlYs5DqkfHfDD4/8A7U3/AAWF+MXxT1f4H/G3QP2cPg18LfFF

34L0m7TwPaeKtV8Y3luI/tFzOLxljhhXIMXlEHErBwxUMPRtQ+Dni9/+DlWw8djwr4kPghPgW+kN

4hGmTf2Ut7/axk+ym62+V52z5vL3btvOMV4R+yZ8Svin/wAEKfib8Y/hb4p/Z4+Ovxk+G/i/xpf+

N/A/ij4XeHf+EiAgvdnmWd3D5iG2eMooy7bnbzCqlNrlqSkqUqu/JU20vKNXlje2l/ZqTS6tXs2V

Ui4OoqW3NT87RdNOVr/9PGk7apNrRXPfvi/8JP20f2mv2LvCXgXX7f4TaN4mn1a6074lXFh4tv8A

Qv8AhKNHhl2wjTLq3srt7QX0Q/fM8ayRqXVFQyfuvqT9k3w74t8C/DC18N+I/h/8OvhxpnhyGHTt

B0rwd4nudcs4rOOMKqkz6dYmLbgAKqOCOSwPFV/2K/iZ8UPjF8BLLxL8XfAFn8LvFuq3d1KnhiHU

U1GXSrLzWFss86Eo87RBXfaFAL4KqQVHrFatODkpLWVm9t0vLTTXRe6m5cqV2Z+7KzjtG6Xpd9++

mu8ko3bsFFFFQUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH/9k=”/>

<OrdQty Qty="1000"/>

</Order>

</FIXML>

## Alternative 2 – Field references implemented as elements

The following is an example of a FIXML file with the field reference EncIssr included in the Instrmt component of an Order as an element.

<?xml version="1.0" encoding="UTF-8"?>

<FIXML xmlns="http://www.fixprotocol.io/2004/fixml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" fixVersion="FIX.5.0SP2" fixExtensionpack="208">

<Order ID="123456" Side="2" TxnTm="2001-09-11T09:30:47-05:00" Typ="2" Px="93.25" Acct="26522154">

<Instrmt Sym="IBM" ID="459200101" Src="1">

<EncIssr>

/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcG

BwcICQsJCAgKCAcHCg0KCgsMDAwMBwkODw0MDgsMDAz/2wBDAQICAgMDAwYDAwYMCAcIDAwMDAwM

DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAARCAApAGEDASIA

AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA

AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3

ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm

p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA

AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx

BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK

U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3

uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9/KKK

/PDSv+DjDwj4z8ReJrLwd+zL+2R8QrTwnrl34evtV8KfDqHVtON3bSFJUWaK8Iz0OGw21lJUZpcy

5uTra/yVl+q+8dny83Tb8/8AJn6H0V5f+yF+07/w1z8G4fGP/Cvfih8MfOu5rX+w/H+g/wBi6zH5

ZA8xrfe+I2zlW3cgGvUKqUXF2ZMZJq6CiiikMKKKKACiiigAorgvjr8ALH9oKw0yx1bxH480XStP

ne4mtfDHiW78PPqLFCqeddWTxXYRMlgkc6IxPzhwFA8D/wCCOvjbxP4z+DHxKXU/E+u+NfBOi/Er

XNJ+H2va1dyX97qOgwSIkZa7kLSXaJci6jSeRnd0jXLtgUQ96UodUub5JxX33louqTd9LBP3YqXR

u3zak/utF697K3U+uaKKKACvw+/4I6f8FQv+GN/BPx08K/8ADO/7UnxV+0fGXxLqH9sfDvwF/bml

R75o18hp/PjxMuzLJt4Doc81+4NfjF/wS/8A24PF3/BM/wANfGTwT4x/ZN/bI8U3eu/FbxB4ksdR

8KfDCa+064tLiVFiZZZZYSxPlFsqpUqykMc1FN2xD1tenJevv0nb8L/IdRXorS9pxfp7lRX/ABt8

z9bfgD8Xv+F+fBrw94x/4Rfxh4L/AOEgtBdf2H4q03+ztZ03JI8u5t9zeVJxnbuPBFefftv/APDS

n/CNaF/wzb/wo7+2PtUn9tf8LK/tT7N5G0eX9n+wfN5m7O7fxjGOa2/2Of2qP+GwvhCfF3/CuPir

8LcX8tj/AGL8QvD/APYmrnywh87yPMf902/Ctu5KtxxWJ+2/+xD/AMNt+GtC03/hb3xx+EX9hXUl

z9q+Gvir+wLnUN6hfLuH8qTzEXGVXAwSTV4mLv7qtqnv0un+X37CoSVtXfdbddV+Z85R/wDD0vzF

3f8ADAW3PzY/4S7OPavvu33+Qnm7fM2jfs+7nvj2r4Ej/wCCB3lyK3/Daf7fpwc4Pxe4P1/0Svvu

3i+zwIm532KF3Ocs2O5PrWl1yJdb/wCXUnXmufl58X/2n/2u/jj/AMFkfjF+zr8G/G/hXwh4M0jw

7omqSeJNZ0S11GTwPFLAGmls7baj3tzPK6BVuJGhRUfhSykcV8Nv2k/299O/bn8ffsbP8Rvhr4w8

TwaZa+K7T4y6n4bgsbnw/ocwEUrpo9vi3uLoTOghjc7VZJPNaRJFMX0J+zF8HfF2gf8ABwR+0x4z

vvC3iOy8H694E8O2ema7Ppk0em6jPEkfmxQ3BURyOmDuVWJXHIFN+G/wd8XWP/ByF8RvHM3hXxHD

4JvvgzY6XbeIH0yZdKuLtb+N2tkuSvlNKFBYoG3ADOKwwUbwoKevOqnNf+77WUdd1rCKumm07bNW

rFSadVx+z7Ll+apKXr8Ur3urq+9zI/4J/ftRfHz4M/8ABT3xx+yl8fvH+l/GOb/hD4fH/hPxtbeG

rfw/cy2xljtprOe1tv3ICybypG5vlYs5DqkfHfDD4/8A7U3/AAWF+MXxT1f4H/G3QP2cPg18LfFF

34L0m7TwPaeKtV8Y3luI/tFzOLxljhhXIMXlEHErBwxUMPRtQ+Dni9/+DlWw8djwr4kPghPgW+kN

4hGmTf2Ut7/axk+ym62+V52z5vL3btvOMV4R+yZ8Svin/wAEKfib8Y/hb4p/Z4+Ovxk+G/i/xpf+

N/A/ij4XeHf+EiAgvdnmWd3D5iG2eMooy7bnbzCqlNrlqSkqUqu/JU20vKNXlje2l/ZqTS6tXs2V

Ui4OoqW3NT87RdNOVr/9PGk7apNrRXPfvi/8JP20f2mv2LvCXgXX7f4TaN4mn1a6074lXFh4tv8A

Qv8AhKNHhl2wjTLq3srt7QX0Q/fM8ayRqXVFQyfuvqT9k3w74t8C/DC18N+I/h/8OvhxpnhyGHTt

B0rwd4nudcs4rOOMKqkz6dYmLbgAKqOCOSwPFV/2K/iZ8UPjF8BLLxL8XfAFn8LvFuq3d1KnhiHU

U1GXSrLzWFss86Eo87RBXfaFAL4KqQVHrFatODkpLWVm9t0vLTTXRe6m5cqV2Z+7KzjtG6Xpd9++

mu8ko3bsFFFFQUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH/9k=</EncIssr>

</Instrmt>

<OrdQty Qty="1000"/>

</Order>

</FIXML>

# Appendix B – Compliance Strategy

The FIXML Schemas are validated using both Apache and Altova software products. Further testing should be done by member firms using various language bindings, such as JAXB, JOBX, .Net, etc.